Welcome

Workshop Materials: bit.ly/3TEjKdd

Hands-on Exercise:

You will use one of the following chatbots depending on the first letter of your last name. Please ensure you have access:

- (A – H) – OpenAI's ChatGPT
- (I – Q) – Microsoft Copilot in Bing
- (R – Z) – Google Gemini

* If you have trouble accessing your assigned chatbot. Feel free to use a different one. Links to these chatbots can be found in the workshop materials.

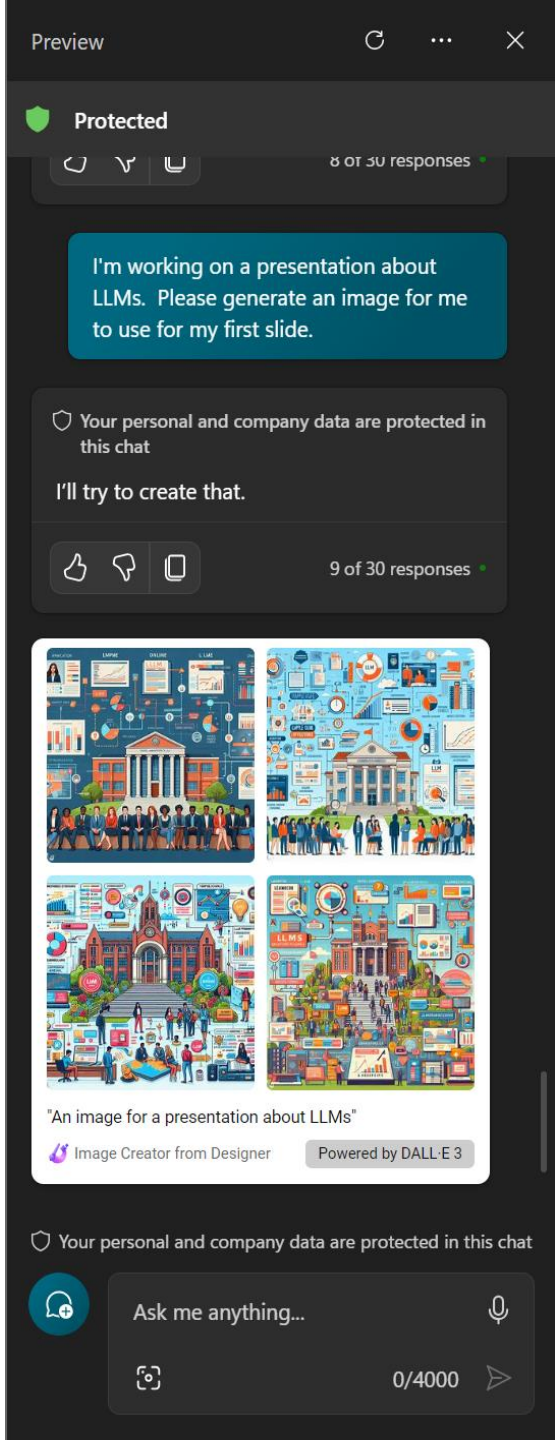
Northwestern
INFORMATION TECHNOLOGY
RESEARCH COMPUTING AND DATA SERVICES

How to Code with LLMs

efrén cruz cortés, *Data Scientist*
Aaron Geller, *Sr. Data Visualization Specialist*
Emilio Lehoucq, *Data Scientist*

April 2, 2024

CoDEx



This Workshop is brought to you by

Northwestern IT
Research Computing and Data Services

Do you have a programming, data, machine learning, statistics, or visualization question about your research?

We're here to help.

- Come find our table in the CoDEx lobby and talk to a consultant today.
- Go to bit.ly/rcdsconsult to request a FREE consultation.

What's in this workshop?

- Short background on LLMs and warnings for their use
- Tips and tricks for better prompting and writing code with LLMs
- Hands-on exercise
- Further Resources

^ Response from Copilot for "Please generate an image showing people attending a workshop about large language models."

Images like the one on the right were generated using DALL-E 3 via Microsoft Copilot and will appear throughout the workshop with a caption below showing the prompt we used.

Background

What is GAI and LLM?

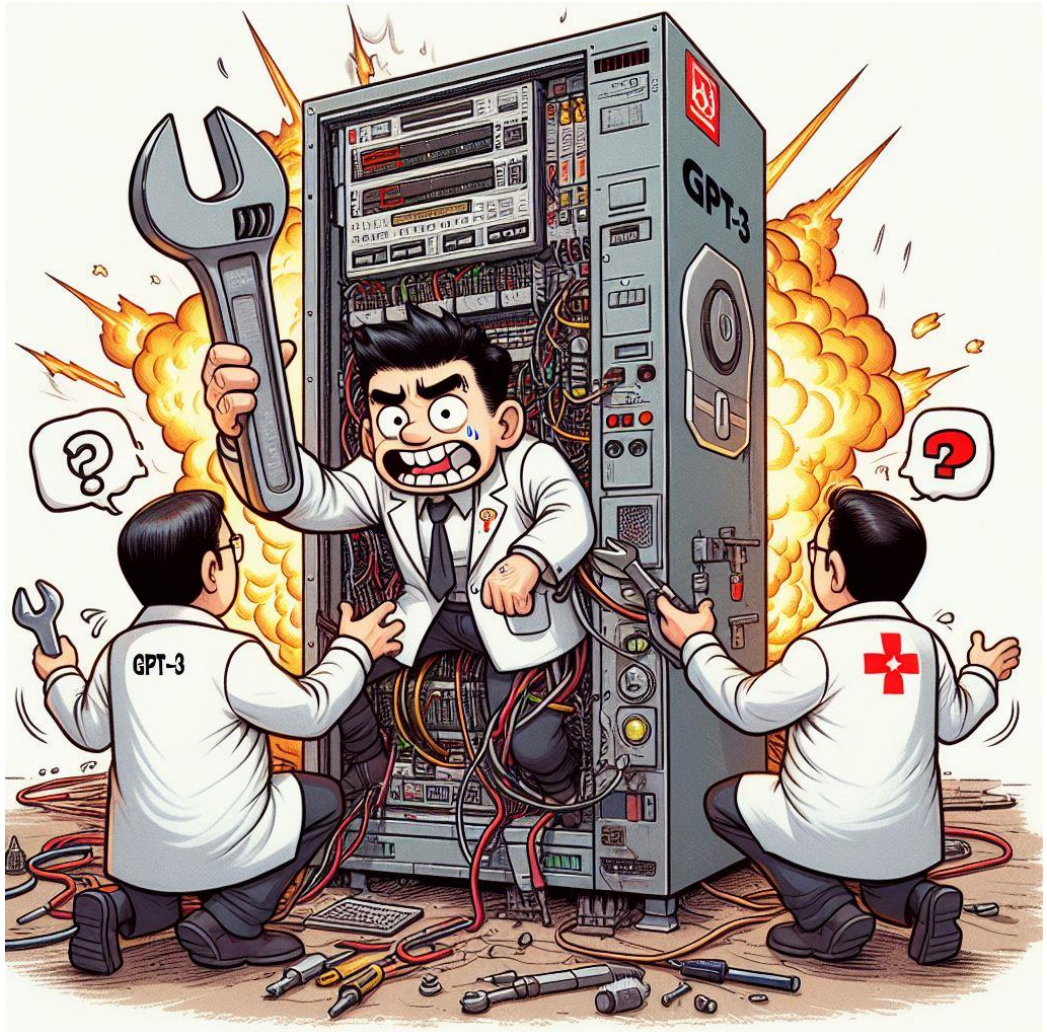
How can an LLM help people to code?

- A Large Language Model (LLM) is a type of generative AI (GAI) tool that interprets text by **predicting the next word**.
  - Is this how you code?
- LLMs are typically **trained on data from the internet** (can have millions – trillions of parameters).
  - Hmm, it's learning from the internet... I wonder what could go wrong...
- Examples :
  - OpenAI's ChatGPT
  - Microsoft's Bing Copilot (also powered by OpenAI's GPT)
  - Google's Gemini
  - LLaMA, Hugging Face, poe.com, claude.ai, etc.
  - Can use some inside IDEs (e.g., GitHub Copilot in VS Code or R Studio; some cost money)

^ Response from Copilot for "Please generate an image of a large language model helping people to code."

Perils of Coding with an LLM

- **Bias**
  - Imai et al. (2022) shows gender bias increases (even for experts) after using an LLM to assist in an analysis task.
  - Don't blindly trust LLMs.
- **Intellectual Property**
  - If you use code produced from an LLM verbatim, do you own it?
  - If you share your code with an LLM, others may see/use it.
- **(Lack of) Privacy**
  - Do not share any data with an LLM that you would not also share with the public on the internet.
- **Cybersecurity**
  - Do not use code verbatim from an LLM without thorough testing and understanding.
- **Errors**
  - An LLM may suggest using erroneous attributes/ functions/ methods/libraries or other bugs. Always test the code!



^ Response from Copilot for "Please generate an image showing the perils of using a large language model."

Northwestern Guidance

https://www.it.northwestern.edu/about/policies/guidance-on-the-use-of-generative-ai.html

Interaction Type	Public Data (Level 1)	Sensitive/Regulated Data (Level 2, Level 3, Level 4)
Conversational/Interactive Mode	Use of publicly available tools (e.g., ChatGPT, Bing Chat, Bard/Gemini, MidJourney, etc.)	Microsoft Copilot for Bing, when signed in with a Northwestern Microsoft account for Level 2 and Level 3 data*
Application Programming Interfaces	Use of Northwestern Azure with OpenAI with appropriate security and access controls	Use of Northwestern Azure + OpenAI with security and access controls that meet or exceed regulatory or data protection requirements

Northwestern's current services posture based on data classification

Northwestern IT suggests using **Microsoft Copilot for Bing.**

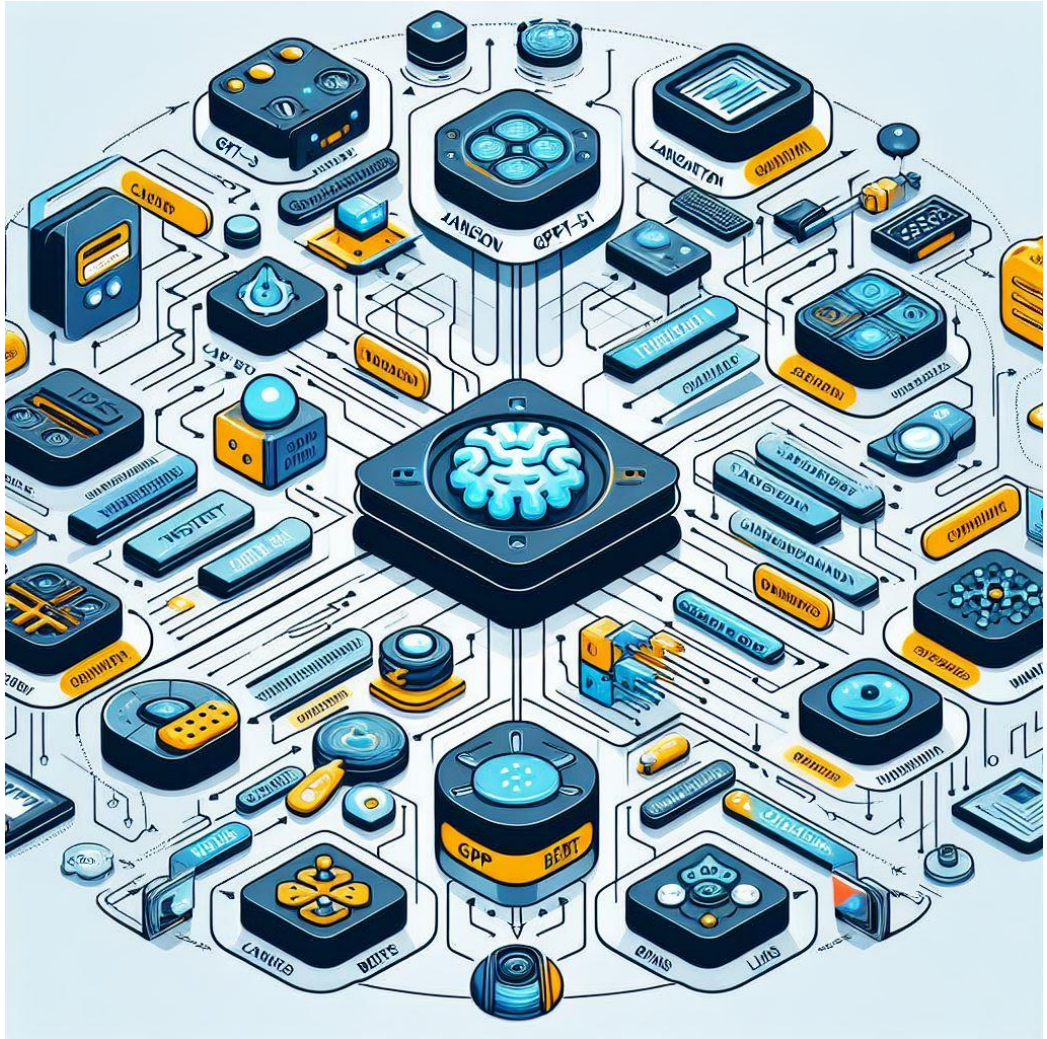
If you have questions about your data security when using GAI, contact security@northwestern.edu

Prompt Engineering

Tips to write effective prompts for coding

What Can You Ask LLMs To Do When Coding?

- Brainstorm ideas, pros/cons
- Initial structure or template for code
- **Generate code**
- Generate unit tests
- Simplify or refactor code
- Debugging
- Explain code that you wrote or found



^ Response from DALL-E 3 for "Please generate an image about the many possible uses of ChatGPT and LLMs."

How To Write Prompts for Coding

Clear, detailed, non-ambiguous, and concise.

DO	***DON'T***
I have 3000 Reddit posts. I want to use topic modeling to find out what the posts are talking about. Before doing the topic modeling, how do I need to prepare the text?	How to pre-process text for NLP?

How To Write Prompts for Coding 2

Tell the LLM what role to take, and your own expertise level.

DO	***DON'T***
You are an expert in computational biology, particularly genomics. What are the key R packages for statistics?	What are the key R packages for statistics?

How To Write Prompts for Coding 3

Describe the programming language, libraries, and other technologies.

DO	***DON'T***
How do I scrape a website in Python using Selenium?	How do I scrape a website in Python?

How To Write Prompts for Coding 4

Explain what the code is for.

DO	***DON'T***
I want to scrape a website that relies heavily on JavaScript. I want to get the source code after loading all elements. How can I do that with Python?	I want to scrape a website. How can I do that with Python?

How To Write Prompts for Coding 5

Specify any constraints or requirements.

DO	***DON'T***
Using ggplot2 in R, I want to iterate over 100 columns, creating the same plot with each. Please write code to do that considering that I'm running the code in my university's high-performance computing cluster.	Using ggplot2 in R, I want to iterate over 100 columns, creating the same plot with each. Please write code to do that.

How To Write Prompts for Coding 6

Be specific about what you want the code to do.

DO	***DON'T***
Write a function in Python that takes two numbers, adds them, and returns the result. The function should raise a type error if the inputs are not numbers.	Write a function in Python that takes two numbers, adds them, and returns the result.

How To Write Prompts for Coding 7

Provide examples of the desired behavior.

DO	***DON'T***
Write a regular expression to match salaries in job postings. These are some examples: String: Minimum $60,000 ANNUAL (12 months) Match: $60,000 String: Rate of Pay $78,035 - $106,517 Match: $78,035 and $106,517 String: starting salary will be from £46,047 up to £61,823 Match: £46,047 and £61,823	Write a regular expression to match salaries in job postings.

How To Write Prompts for Coding 8

Ask the LLM to explain itself and any assumptions.

DO	***DON'T***
Write a regular expression to match salaries in job postings. Explain any assumptions that you're making about the data and what the match should be.	Write a regular expression to match salaries in job postings.

How To Write Prompts for Coding 9

Ask the LLM to work step-by-step.

DO	***DON'T***
What is this code doing? iris %>% group_by(Species) %>% summarize_all(mean) %>% ungroup %>% gather(measure, value, -Species) %>% arrange(value) Work step-by-step.	What is this code doing? iris %>% group_by(Species) %>% summarize_all(mean) %>% ungroup %>% gather(measure, value, -Species) %>% arrange(value)

How To Write Prompts for Coding 10

Specify how you want the LLM to give you the information.

DO	***DON'T***
What are different packages for supervised learning in R, as well as their pros and cons? Provide the information in a table with only one row per package.	What are different packages for supervised learning in R, as well as their pros and cons?

How To Write Prompts for Coding 11

Reformulate prompts as needed!

Recap

- Clear, detailed, non-ambiguous, and concise.
- Tell the LLM what role to take, and your own expertise level.
- Describe the programming language, libraries, and other technologies.
- Explain what the code is for.
- Specify any constraints or requirements.
- Be specific about what you want the code to do.
- Provide examples of the desired behavior.
- Ask the LLM to explain itself and any assumptions.
- Ask the LLM to work step-by-step.
- Specify how you want the LLM to give you the information.
- Reformulate prompts as needed!

How To Use Conversations For Coding

- LLMs are sensitive to chat history.
  - Treat interactions as conversations.
  - Use different conversations for different questions.
- Same prompt can give different results across conversations.
- You can copy code from one conversation to another to get feedback.
- Avoid long conversations.

^ Response from DALL-E 3 for "Please generate an image of a person having multiple conversations with a computer"

Example conversation

You're an expert data scientist. Explain for a beginner.

I have a list of one billion numbers. As a preprocessing step, I want to first sort them in ascending order and then divide each of them by the previous number (the first number doesn't need to be divided).

For example, if the list is [3, 2, 4, 6], the correct order is [2, 3, 4, 6], and the output I want is [2, 1.5, 1.33, 1.5].

What are some ways in which I can do this in Python both with built-in functions and external libraries? What are their pros and cons (including complexity of syntax, readability, efficiency, and dependency on external libraries)?

Output only a table with as many rows as there are approaches and five columns: approach, explanation, pros, cons, and assumptions that you're making.

ChatGPT

Approach	Explanation	Pros	Cons	Assumptions
Built-in Python	Utilize built-in functions `sorted()` and list comprehensions. First, sort the list using `sorted()`, then iterate through the sorted list, dividing each element by its previous element.	- No external dependencies	- Moderate readability	Assumes the list can fit into memory comfortably.
NumPy	Use NumPy library for efficient array operations. Convert the list to a NumPy array, then sort it using `numpy.sort()`. Next, compute element-wise division using array slicing to avoid looping, which is inherently faster in NumPy.	- High efficiency	- Requires NumPy library	Assumes NumPy is installed and familiarity with array slicing operations.
Pandas	Convert the list to a Pandas DataFrame, then use its built-in sorting and shifting capabilities. First, create a DataFrame from the list, sort it using `DataFrame.sort_values()`, and compute the desired output using `shift()` method.	- Familiar DataFrame operations	- Requires Pandas library	Assumes Pandas is installed and familiarity with DataFrame manipulation.

Parallelization	Employ parallel processing to handle the sorting and division concurrently. Divide the list into chunks, sort each chunk independently, then perform division in parallel.	- Potential for significant speedup with large data	- Complexity in managing parallel processes	Assumes access to parallel processing libraries and familiarity with parallel programming concepts.
External Sort	Use an external sorting algorithm, like merge sort or external quicksort, suitable for large datasets that cannot fit into memory. After sorting externally, perform the division as usual.	- Suitable for datasets too large for memory	- Complexity in implementation	Assumes access to external sorting algorithms or libraries and familiarity with their implementation.

Each approach has its own trade-offs in terms of efficiency, readability, and dependencies. The choice depends on the specific requirements of the task, such as the size of the data, available computational resources, and familiarity with external libraries.

Anonymous

How do I implement the first option--built-in python? Work step-by-step, explain, and annotate the code.

ChatGPT

Sure, let's break down the implementation of the first option using built-in Python functions step-by-step:

python Copy code

```python
# Define a function to perform the desired operation
def preprocess_list(numbers):
    # Sort the list in ascending order
    sorted_numbers = sorted(numbers)

    # Initialize a list to store the results of division
    result = []

    # Iterate through the sorted list
    for i in range(len(sorted_numbers)):
        # If it's the first number, append it to the result list
        if i == 0:
            result.append(sorted_numbers[i])
        else:
            # Divide the current number by the previous number and append the result
            result.append(sorted_numbers[i] / sorted_numbers[i-1])

    # Return the preprocessed list
    return result
```

```
# Example usage
input_list = [3, 2, 4, 6]
output_list = preprocess_list(input_list)
print(output_list)  # Output: [2, 1.5, 1.3333333333333333, 1.5]
```

Explanation of the code:

1. We define a function `preprocess_list()` that takes a list of numbers as input.
2. Inside the function, we sort the input list in ascending order using the `sorted()` function and assign it to `sorted_numbers`.
3. We initialize an empty list `result` to store the preprocessed numbers.
4. We iterate through the sorted list using a `for` loop, where `i` represents the index of each element.
5. For each element in the sorted list:
   - If it's the first number (`i == 0`), we simply append it to the `result` list.
   - Otherwise, we divide the current number by the previous number (`sorted_numbers[i-1]`) and append the result to the `result` list.
6. Finally, we return the preprocessed list.
7. We demonstrate the usage of the function with an example input list `[3, 2, 4, 6]`, and print the output list.

This implementation is straightforward and doesn't require any external libraries. However, it may not be the most efficient solution for very large datasets, as it involves sorting the entire list and iterating through it sequentially.

Anonymous

Provide assert statements for the function. Include core and edge cases.

ChatGPT

Sure, let's add some `assert` statements to validate the function for core and edge cases:

python Copy code

```python
def preprocess_list(numbers):
    # Sort the list in ascending order
    sorted_numbers = sorted(numbers)

    # Initialize a list to store the results of division
    result = []

    # Iterate through the sorted list
    for i in range(len(sorted_numbers)):
        # If it's the first number, append it to the result list
        if i == 0:
            result.append(sorted_numbers[i])
        else:
            # Divide the current number by the previous number and append the result
            result.append(sorted_numbers[i] / sorted_numbers[i-1])

    # Return the preprocessed list
    return result

# Test cases
# Core case: Basic example
assert preprocess_list([3, 2, 4, 6]) == [2, 1.5, 1.3333333333333333, 1.5]
```

```
# Edge case: Empty list
assert preprocess_list([]) == []

# Core case: List with one element
assert preprocess_list([5]) == [5]

# Core case: List with negative numbers
assert preprocess_list([-3, -2, -4, -6]) == [-6, 0.6666666666666666, 2.0, 1.5]

# Core case: List with duplicate elements
assert preprocess_list([5, 5, 5, 5]) == [5, 1.0, 1.0, 1.0]

print("All assertions passed!")
```

Explanation of the test cases:

1. Core case: Basic example - Verifies that the function produces the correct output for a simple input list **`[3, 2, 4, 6]`**.
2. Edge case: Empty list - Checks if the function handles an empty input list correctly by returning an empty list.
3. Core case: List with one element - Tests the function with a single-element list to ensure it returns the same element.
4. Core case: List with negative numbers - Ensures the function handles negative numbers properly by providing a list **`[-3, -2, -4, -6]`**.
5. Core case: List with duplicate elements - Checks if the function correctly handles lists with duplicate elements by providing **`[5, 5, 5, 5]`**.

If all assertions pass, it indicates that the function behaves as expected for the given test cases.

Useful Features of Major Chatbots

ChatGPT 3.5	Microsoft Bing Copilot	Google Gemini
• Saves your past conversations, so that you can go back to them. • Allows you to share a public link to a conversation.	• Provides different conversation styles for more creativity or precision. • Sometimes provides URLs where you can potentially go read more about the answer. • Allows you to upload images.	• Allows you to Google an answer, so that you can potentially read more about it. • Allows you to upload images. • Saves your past conversations, so that you can go back to them. • Allows you to share a public link to a conversation.

Exercises

Your turn to work with LLMs

Welcome

Workshop Materials: bit.ly/3TEjKdd

Hands-on Exercise:

You will use one of the following chatbots depending on the first letter of your last name. Please ensure you have access:

- (A – H) – OpenAI's ChatGPT
- (I – Q) – Microsoft Copilot in Bing
- (R – Z) – Google Gemini

* If you have trouble accessing your assigned chatbot. Feel free to use a different one. Links to these chatbots can be found in the workshop materials.

Exercise 1 - Food Research  

- Use an LLM to generate code to determine ***which country consumed the least meat per capita in 2020***.
  - Use the data here. Link also found in the GitHub page with the workshop materials.
  - Run the code yourself to make sure it works.
  - More information on the data is found on the GitHub page.

Directives

- If you don't know how to download the data – ask your LLM!
- You can use any programming language of your choice.
- Try solving it only with prompt engineering and not by coding the solution yourself.

Exercise 1 - Food Research 🥗😋

This is what the columns mean:

- LOCATION is the country.
- SUBJECT is meat type (beef, pig, poultry, sheep).
- MEASURE is the metric reported.
  - KG_CAP - kilograms per capita.
  - THND_TONNE - thousand tonnes.
- TIME is the year.
- VALUE is the value for the indicator.

Exercise 2 – Food Research 2.0 

Was exercise 1 too easy? Try this!

*Which country has the least **annual meat consumption per capita** the most years between 2000 and 2020?*

For example, if country X consumed the least for 7 yrs, and country Y consumed the least for 13 years. Country Y would be the answer.

Exercise Recap 

- The correct answer for part 1 is: _____
- The correct answer for part 2 is: _____

Exercise Recap 

- The correct answer for part 1 is: Ethiopia
- The correct answer for part 2 is: _____

Exercise Recap 

- The correct answer for part 1 is: Ethiopia
- The correct answer for part 2 is: India

Exercise Recap 

- Who was able to solve it?
- How many per group?

Exercise Recap 

- If you solved it, what prompting strategies worked?
- If you couldn’t solve it on time, what was causing trouble?

Exercise Recap 

- Did you get erroneous information from the LLM?
- What types of errors did you encounter/identify from the LLMs output?

Exercise Recap 

- Was keeping the context of the problem in one conversation useful?
- If you had to change conversation to reset context history, how did that help?

Exercise Recap 

- How did you help the LLM?
  - Did you break down the problem into simpler steps?
  - Did you ask the LLM to explain its reasoning?
  - Did you provide examples of what you wanted it to do?

Questions?

Feedback Sheets

- Please don’t forget to fill out the feedback sheets you found on your seat!
- Drop it off on your way out.

Thank you!

References

References for ChatGPT Advice and Tips

- OpenAI's Prompt Engineering Guide
- Andrew Ng's Prompt Engineering Course
- Google's Prompt Engineering Article

Other

- https://www.zdnet.com/article/how-to-use-chatgpt-to-write-code/
- https://www.zdnet.com/article/bard-vs-chatgpt-can-bard-help-you-code/
- https://www.pluralsight.com/blog/software-development/how-use-chatgpt-programming-coding
- https://www.wikihow.com/Get-Chatgpt-to-Write-Code
- https://www.griproom.com/fun/how-and-why-to-use-chat-gpt-to-generate-code-snippets
- https://bdtechtalks.com/2023/06/26/chatgpt-coding-tips/
- https://www.fiverr.com/resources/guides/programming-tech/write-code-with-chatgpt
- https://stratoflow.com/can-chatgpt-write-code/
- https://www.nature.com/articles/d41586-023-01833-0
- https://arxiv.org/abs/2304.11938
- https://arxiv.org/abs/2301.08653
- https://arxiv.org/abs/2307.12596
- https://arxiv.org/abs/2308.02828
- https://arxiv.org/abs/2308.02312
- https://arstechnica.com/information-technology/2023/09/telling-ai-model-to-take-a-deep-breath-causes-math-scores-to-soar-in-study/

References for GitHub Copilot

- https://docs.github.com/en/copilot/getting-started-with-github-copilot?tool=vimneovim
- https://docs.github.com/en/copilot/quickstart
- https://github.com/features/copilot/
- https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/
- https://www.saverilawfirm.com/our-cases/github-copilot-intellectual-property-litigation
- https://github.blog/2023-06-20-how-to-write-better-prompts-for-github-copilot/
- https://dev.to/github/a-beginners-guide-to-prompt-engineering-with-github-copilot-3ibp
- https://www.freecodecamp.org/news/developer-productivity-with-github-copilot/
- https://code.visualstudio.com/docs/editor/github-copilot
- https://medium.com/@rajesh.jadav/using-github-copilot-effectively-8b46604299ed
- https://realpython.com/github-copilot-python/

Extra Materials

What was not in the workshop?

- A technical introduction to LLMs
- A complete error typology of ChatGPT
- A complete discussion of social perils involved with Generative AI
- All possible prompt engineering tips and tricks

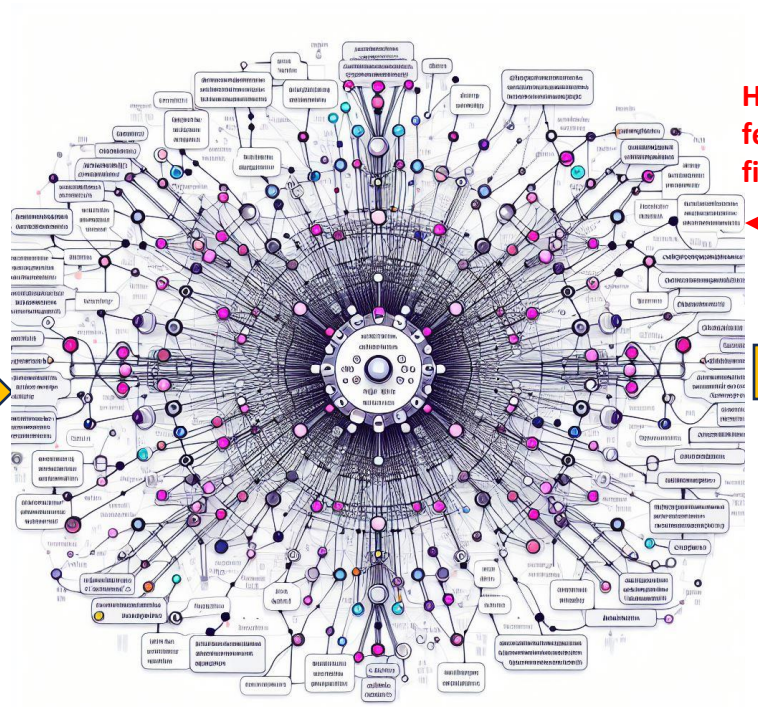
^ Response from DALL-E 3 for "Please generate an image showing good information left on the cutting room floor"

Language Model Creation

Input training data (e.g., from the internet)

^ Response from Copilot for "Please generate an image of training data from the internet for a large language model"

Neural network (transformer), to predict the best next word



^ Response from Copilot for "Please generate an image of a neural network used in large language models"

Human feedback + fine tuning

Interpret text from a user to provide responses

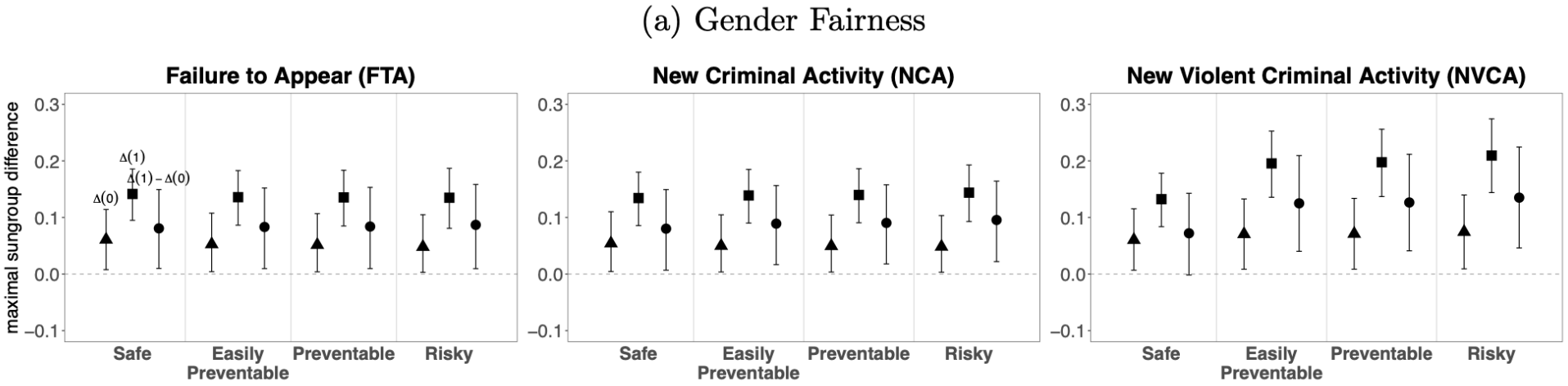
^ Response from Copilot for "Please generate an image of a large language model providing an answer to a person's question on their laptop"

Risks of AI influencing experts counterproductively

Do experts perform better with AI?

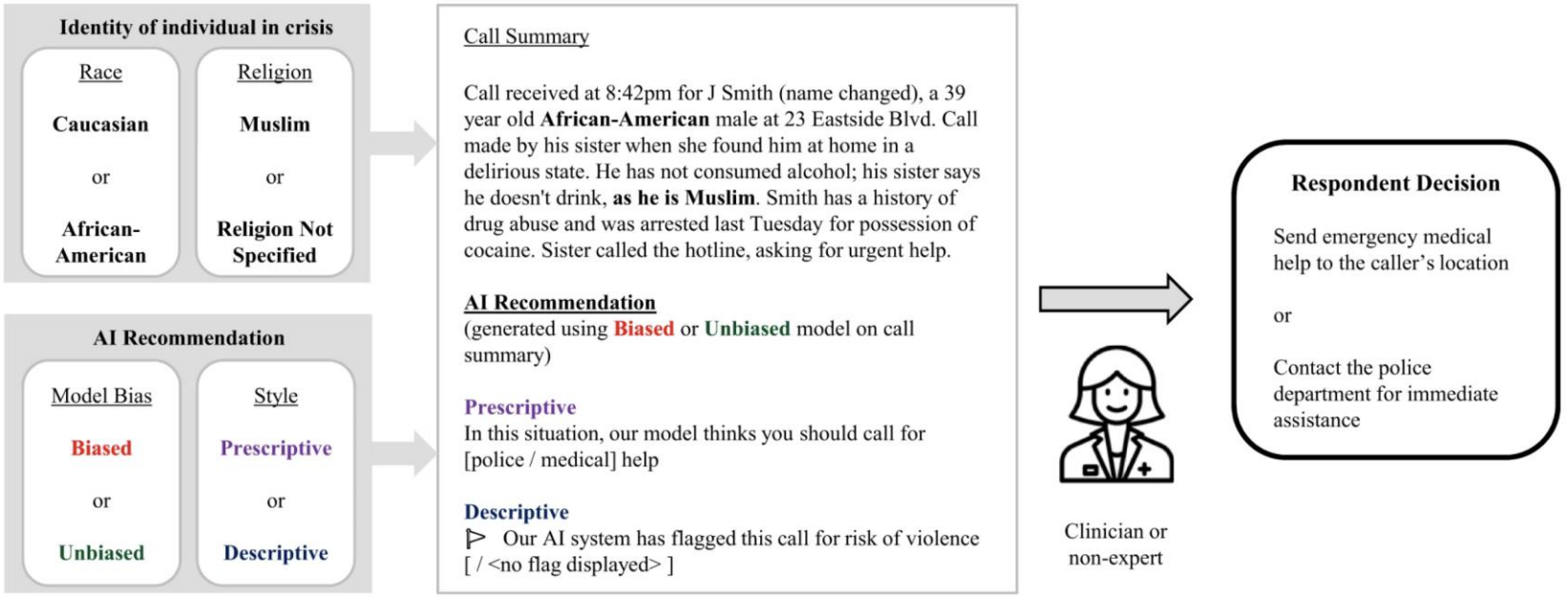
Imai et al. study of 2022 shows judges are influenced by A. I.

- The figure below shows gender bias increase after using an AI assist



Imai et al, Royal Statistical Society, 2022

Do experts perform better with AI?



Results

Participant decisions are unbiased without AI advice. However, both clinicians and non-experts are influenced by prescriptive recommendations from a biased algorithm, choosing police help more often in emergencies involving African-American or Muslim men. Crucially, using descriptive flags rather than prescriptive recommendations allows respondents to retain their original, unbiased decision-making.

Adam et al. Nature Comm. Med. 2022

Do experts perform better with AI?

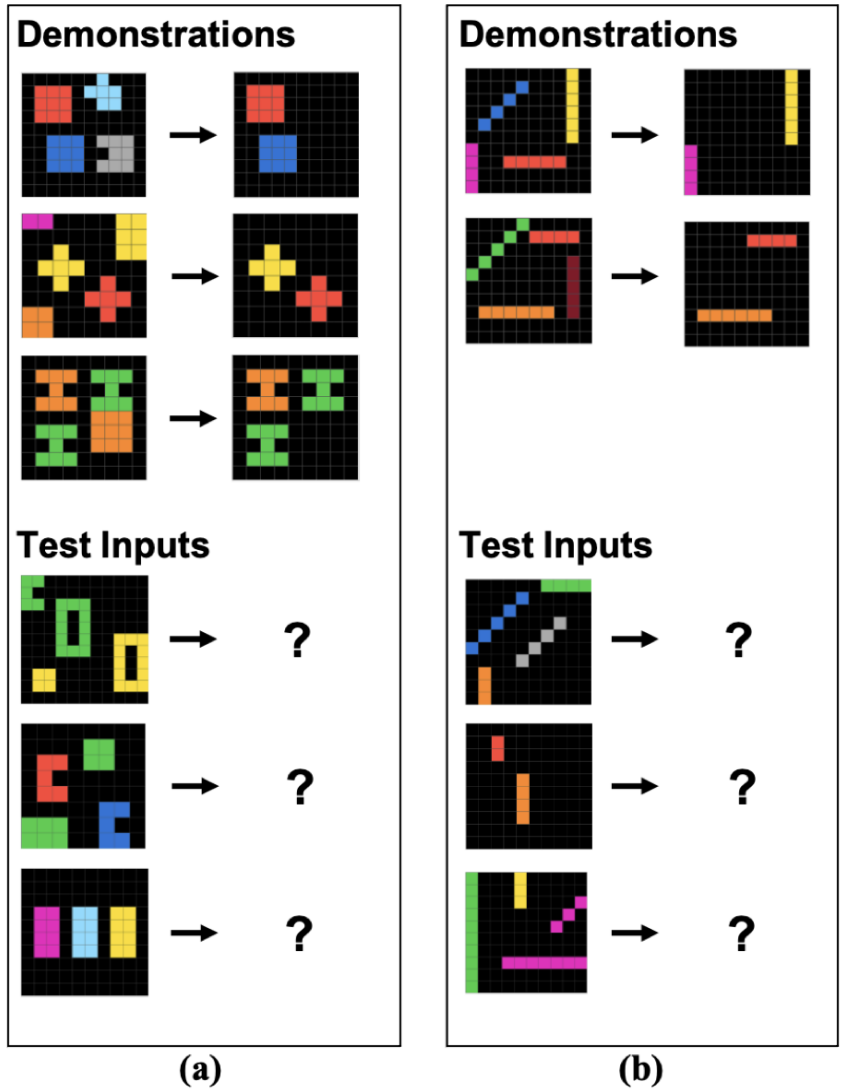
Similar results have been observed for medical diagnosis (Wiens et al. Forthcoming), and for police arrests (Johnson and Johnson, Sci. American, 2023).

So, likely you will also be influenced, sometimes against your better judgement, even if you are an expert in your field!!

Some Systematic Errors

Spatial Reasoning Errors

GPT was tested on spatial/semantic reasoning tasks. It did not perform well.



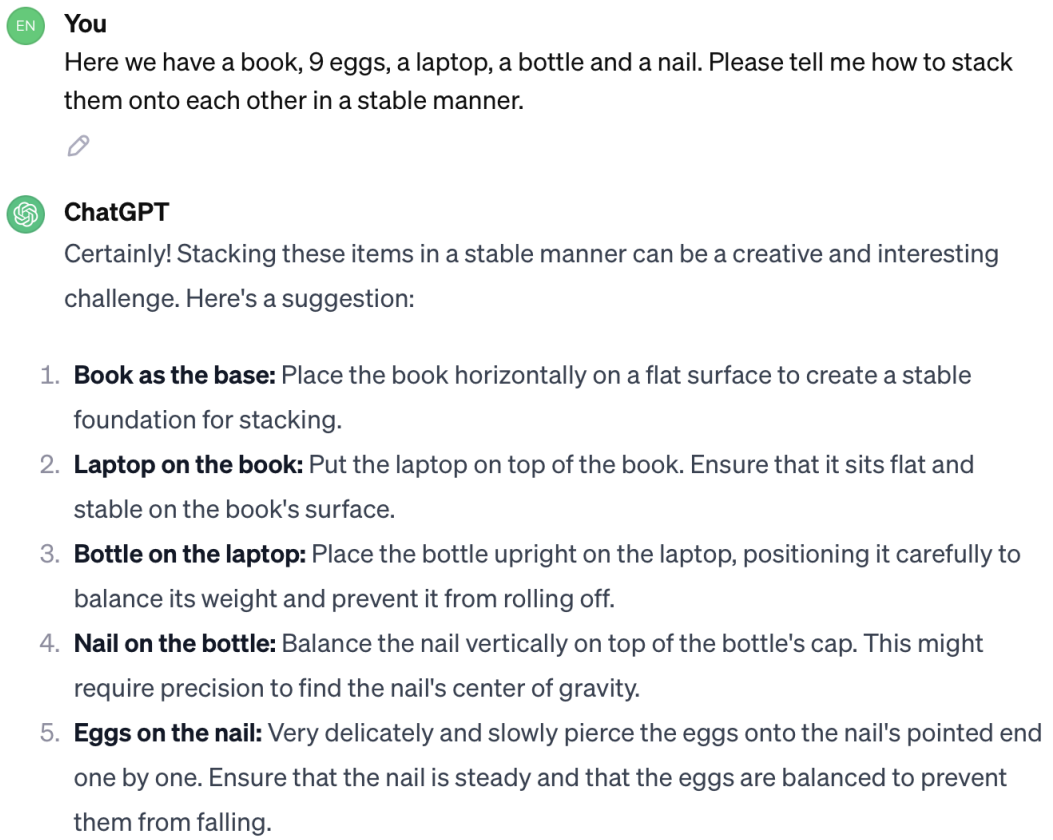
Mitchell et al. 2023. Comparing Humans, GPT-4, and GPT-4V On Abstraction and Reasoning Tasks.

Spatial Reasoning Errors

The famous stacking problem:

GPT-4 is better than ChatGPT, however, it requires constant update and fine-tuning.

If you ask it not to break the eggs, it will stack them one on top of each other.



You

Here we have a book, 9 eggs, a laptop, a bottle and a nail. Please tell me how to stack them onto each other in a stable manner.

ChatGPT

Certainly! Stacking these items in a stable manner can be a creative and interesting challenge. Here's a suggestion:

1. **Book as the base:** Place the book horizontally on a flat surface to create a stable foundation for stacking.
2. **Laptop on the book:** Put the laptop on top of the book. Ensure that it sits flat and stable on the book's surface.
3. **Bottle on the laptop:** Place the bottle upright on the laptop, positioning it carefully to balance its weight and prevent it from rolling off.
4. **Nail on the bottle:** Balance the nail vertically on top of the bottle's cap. This might require precision to find the nail's center of gravity.
5. **Eggs on the nail:** Very delicately and slowly pierce the eggs onto the nail's pointed end one by one. Ensure that the nail is steady and that the eggs are balanced to prevent them from falling.

https://medium.com/@nathanbos/gpt-4-may-have-general-intelligence-but-dont-let-it-near-your-easter-eggs-925f48326d51

Spatial Reasoning Errors

EN **You**

Is it better to balance a chicken egg on a flat counter with its pointed side down, or on its side?

ChatGPT

Balancing a chicken egg with its pointed side down is generally more stable than balancing it on its side. The pointed end of the egg is smaller and more tapered, providing a more secure base for balancing. When the egg is balanced on its side, it has a higher likelihood of rolling or tipping over due to the broader shape.

https://medium.com/@nathanbos/gpt-4-may-have-general-intelligence-but-dont-let-it-near-your-easter-eggs-925f48326d51

Spatial Reasoning Errors

You

to my right is aaron, to my left is emilio, to emilio's left is colby, and next to colby is aaron. who is in front of emilio?

ChatGPT

Let's break down the seating arrangement:

- To your right: Aaron
- To your left: Emilio
- To Emilio's left: Colby
- Next to Colby: Aaron

So, the person in front of Emilio is Colby.

You

To my right is Aaron, to my left is Emilio, to Emilio's left is Colby, and to Colby's left is Aaron. Who is in front of Emilio?

ChatGPT

Based on the information provided, it seems like there might be a circular arrangement. If we assume you're facing forward, and the people are arranged in a circle, then Aaron is in front of Emilio.

Note: One word can make a great difference!!

Semantic Reasoning Errors

Polysemy and change of variables

During a rock-paper-scissors game, I changed the meaning of XYZ from "action" to "outcome". It was not able to:

1. Keep track of two distinct meanings for the same symbols
2. Update appropriately the meaning of a symbol.

1. Here's a table indicating the opponent's move, the strategy instruction, and our move:

Opponent's Move (Rock, Paper, Scissors)	Strategy Instruction (X, Y, Z)	Our Move (Rock, Paper, Scissors with new symbols)
Rock	Y (Draw)	D (Draw with Paper)
Paper	X (Lose)	L (Lose with Rock)
Scissors	Z (Win)	W (Win with Scissors)

1. Based on the new instructions:

- X: You need to lose the round.
- Y: You need to end the round in a draw.
- Z: You need to win the round.

1. Choose new symbols for our moves:

- L: Represents losing the round.
- D: Represents drawing the round.
- W: Represents winning the round.

Semantic Reasoning Errors

Type tracking.

During the same R-P-S investigation. It was not able to transform both sides of an equation using modulo function.

```
if opponent_score == your_score:
    round_score = 3  # Draw
elif (opponent_score + 1) % 3 == your_score:
    round_score = 6  # Win
```

Anatopic Reasoning Errors

Generative AI is particularly bad at taking things out of their usual context. This is called anatopy in literature.

I found many examples, but here is one. Trying to make it think about the most complex or obscure concept without referring to QM is very hard. It is unable to disentangle cliché correlations. Since everyone online claims QM is the most complex, and it doesn't have many examples without this mention, it is hard to move away from it.

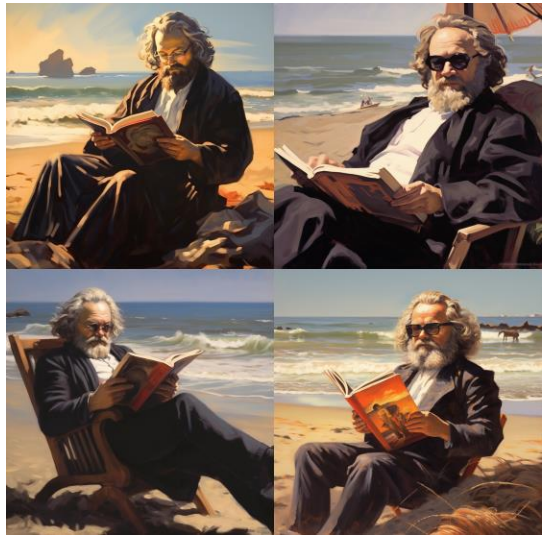
Image models are more obvious. For example Karl Marx reads at the beach with his suit because in all his pictures he's wearing a suit.

EN **You**

Without falling into clichés, what is the most complex concept to understand?

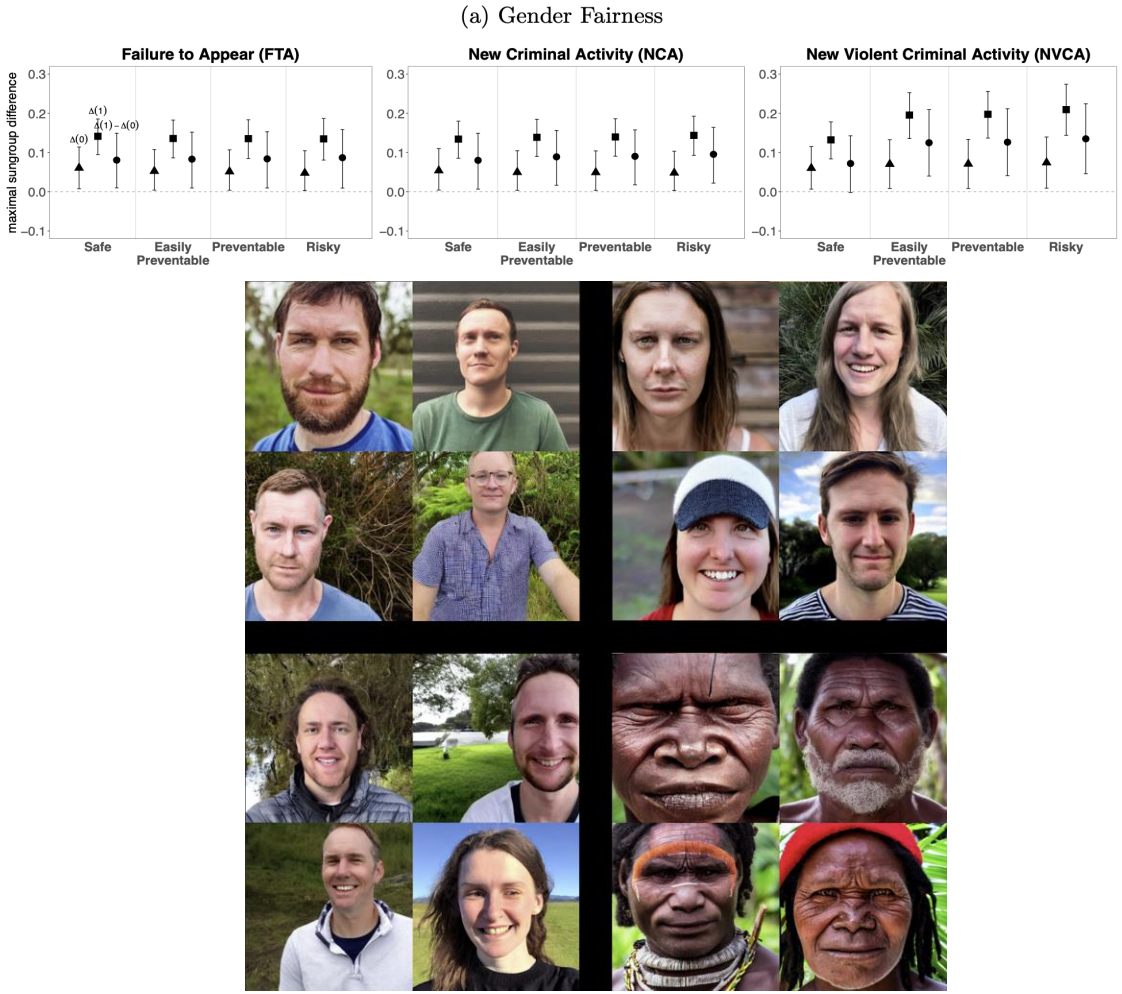
ChatGPT

Determining the single most complex concept can be subjective, as it depends on individual perspectives and areas of expertise. However, one concept often considered challenging is quantum mechanics. Quantum mechanics describes the behavior of particles at the microscopic level and is known for its non-intuitive nature, involving concepts such as superposition, entanglement, and wave-particle duality. Understanding quantum mechanics often requires a departure from classical intuition, making it a complex and fascinating field.



Ethical and Safety Considerations

Bias



- Gender and racial bias may be inherent in GAI responses
- Even human experts may become more biased after using a GAI
- See Imai+2022; Adam+2022, Johnson&Johnson2023, https://sourojitghosh.github.io/publication/paper15

See end of presentation for extra slides with more details

Example of Biased Output

- **“Person” == Light-skinned, Western Man, and Sexualization of Women of Color: Stereotypes in Stable Diffusion** (https://sourojitghosh.github.io/publication/paper15)
  - Clockwise from top left are the results of four prompts to show “a person” from Oceania, Australia, Papua New Guinea and New Zealand. Papua New Guinea, where the population remains mostly Indigenous, is the second most populous country in Oceania. (credit: Ghosh et al./EMNLP 2023 — AI GENERATED IMAGE)

Intellectual Property (IP)

Jason Allen's “Theatre d’Opera Spatial” Midjourney + Photoshop, 1st place at a Colorado State Fair contest.
BUT he cannot copyright it!

- GAI and LLMs are often trained on data from the internet, which can include copyrighted/trademarked text and images and produce "derivative work" from these training data.
- Existing (and presumably future) lawsuits
- Stack Overflow, Reddit, others will charge LLMs for use of their data.
- Think about your IP before feeding your data to an LLM!

Intellectual Property (IP)

- GAI and LLMs are often trained on data from the internet, which can include copyrighted/trademarked text and images and produce "derivative work" from these training data.
- Existing (and presumably future) lawsuits:
  - Visual artists and image owners suing image GAI companies
  - Authors suing LLMs
  - Most legal issues depend on the interpretation of the "fair use doctrine"
- Stack Overflow, Reddit, others will charge LLMs for use of their data.
- Various ways being devised for content creators to "opt out" or even "poison" their works (but would be better for GAI/LLM companies to play nice)
- Think about your IP before feeding your data to an LLM!
- Art generated entirely by AI, without any human input, is not eligible for copyright protection under current US law (though many nuances)

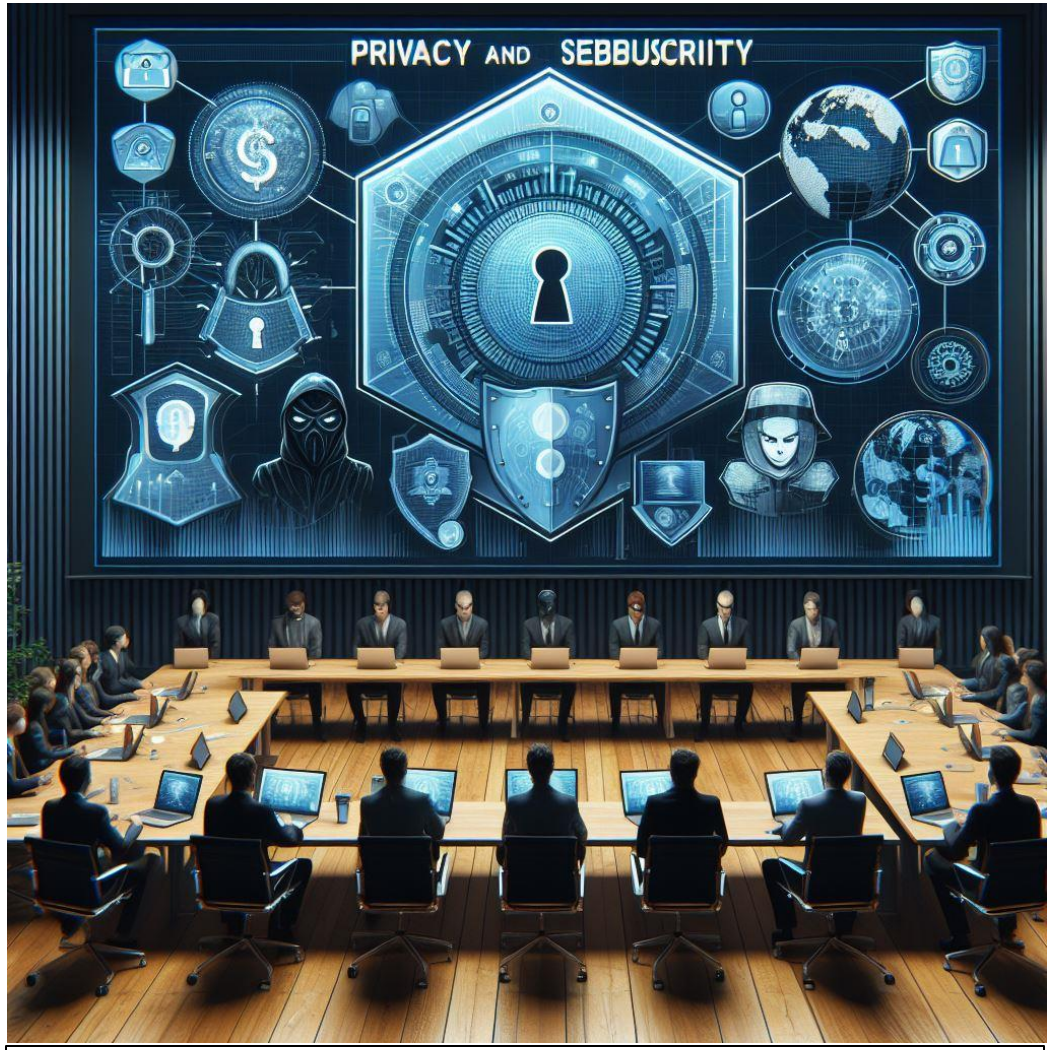
Example of an IP issue

Jason Allen's “Theatre d’Opera Spatial” made with Midjourney, and extensive prompt engineering + manual edits in Photoshop, won 1st place at a Colorado State Fair contest.

BUT he cannot copyright it!

(Lack of) Privacy

- Very quickly evolving landscape!
  - ChatGPT privacy policy has been updated essentially every time I go back to the page (~every 6 months)!
- Personal data, user content, social media interactions, device info, geolocation, etc. can be used to train their models, and can all be shared with 3rd parties.
- "We'll retain your Personal Information for only as long as we need..."
- Since April 25, 2023 you can disable chat history
  - Won't use your conversations to train models
  - Still retain your conversations for 30 days (to monitor for abuses)
  - But you lose the ability to see old conversations



^ Response from DALL-E 3 for "Please generate an image for me to use in a presentation about privacy and cybersecurity issues when using LLMs."

Cybersecurity Risk

- ChatGPT can generate code that is vulnerable to security threats or weaknesses.
- Bad actors can exploit code generated by ChatGPT even if it’s functionally correct.
- This risk depends on what you're using your code for. It's particularly salient if you're writing production code or code that connects to the internet.
- To mitigate this risk, you should treat code generated by ChatGPT as containing vulnerabilities. You should review it, make sure you understand what it's doing, and test it.

^ Response from DALL-E 3 for "Please generate an image about cybersecurity risk when using ChatGPT"

Error Typology

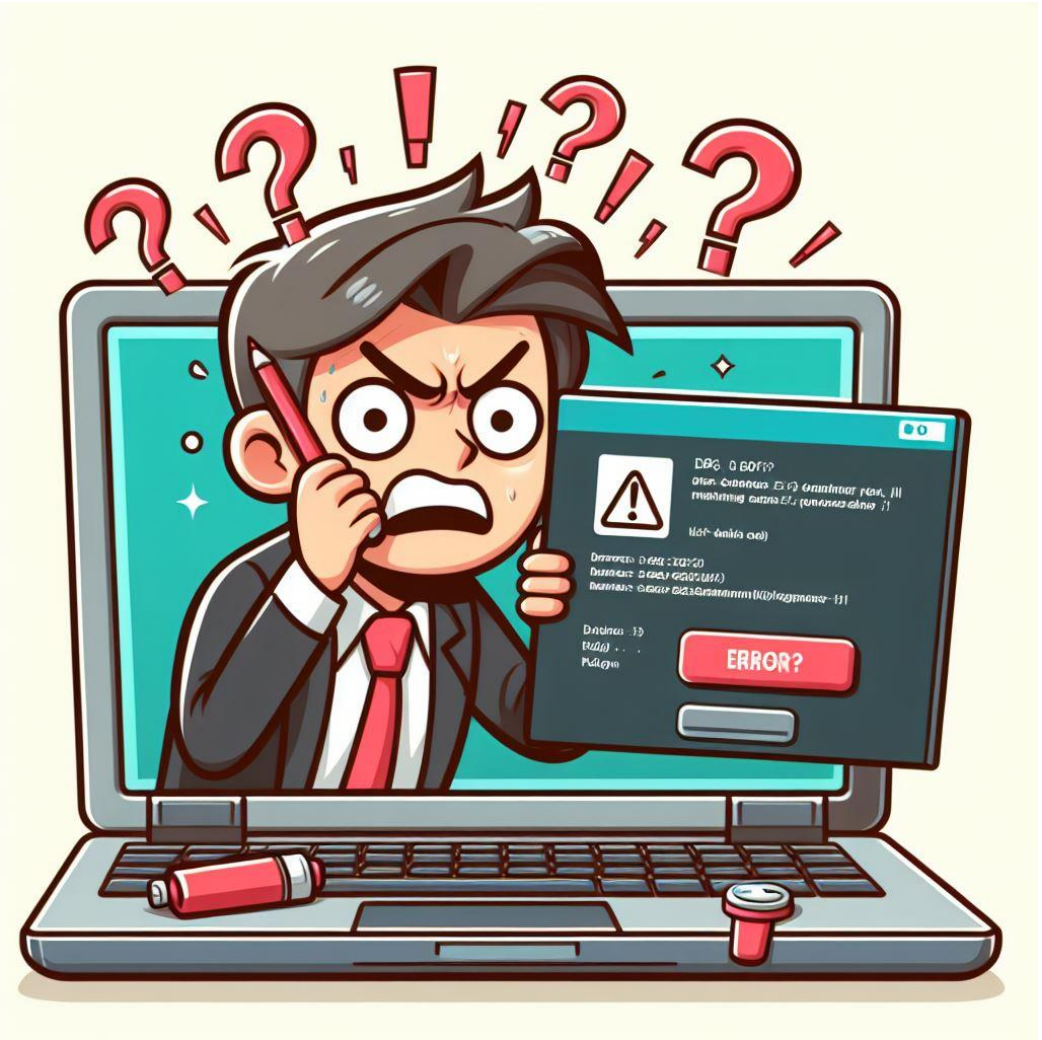
Many errors are obvious: the content is false, even if presented through what seems sound reasoning. However, what is the origin of these errors?

There are a few identified "reasoning" pitfalls LLMs have. We list just three:

- Spatial reasoning
- Semantic reasoning
- Anatopic reasoning

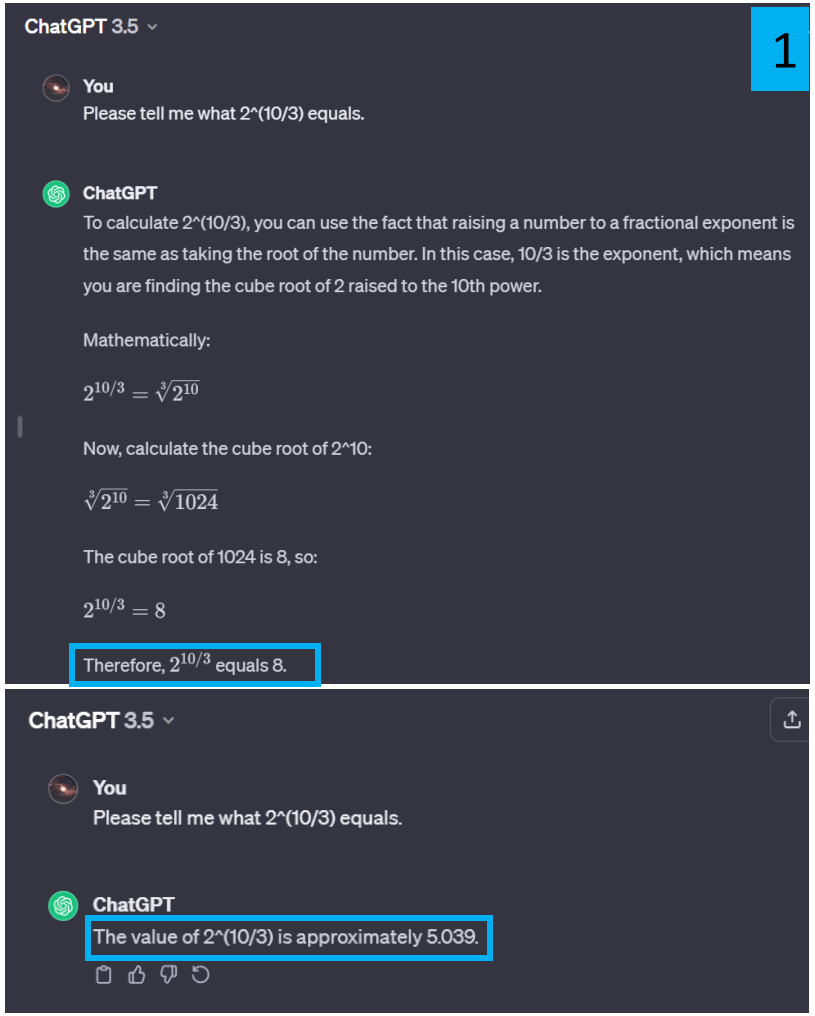
Do not mistake eloquence for truth!

See end of presentation for extra slides with more details



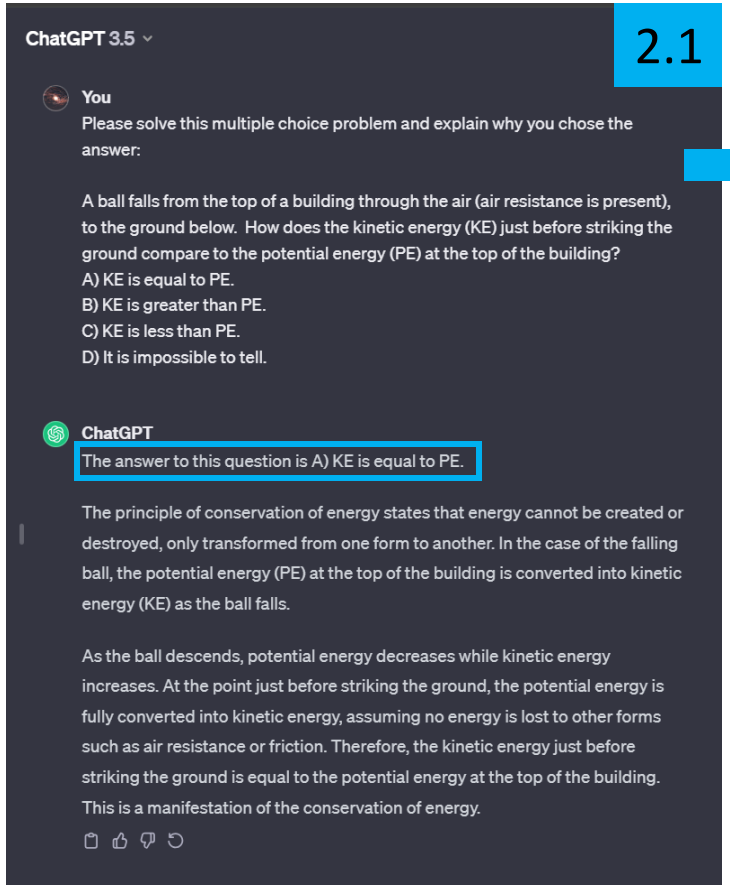
^ Response from DALL-E 3 for "Please generate an image of a person confused by errors displayed on the monitor of their laptop"

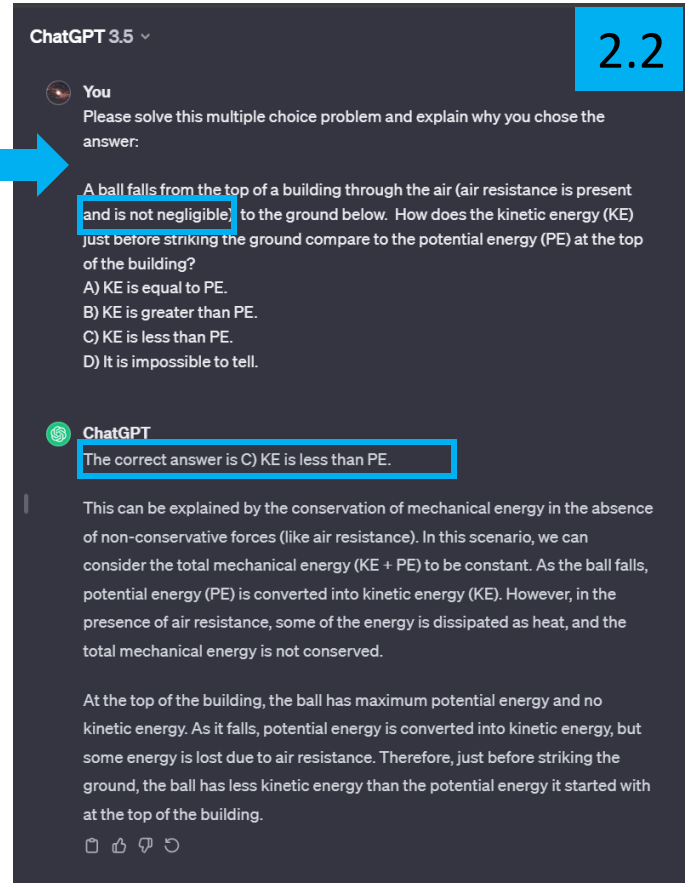
Examples with Errors and Incorrect Output



Don't ask ChatGPT to do math!

2^(10/3) ~ 10.0794





Physics question about conservation of energy with dissipation (air resistance). The exact prompt matters!

Correct answer = C

Errors and Incorrect Output

- Who Answers It Better? An In-Depth Analysis of ChatGPT and Stack Overflow Answers to Software Engineering Questions (Kabir et al. 2023)
  - "a comprehensive analysis of ChatGPT's replies to 517 questions from Stack Overflow" shows that "52% of ChatGPT's answers contain inaccuracies and 77% are verbose. Nevertheless, users still prefer ChatGPT's responses 39.34% of the time due to their comprehensiveness and articulate language style."
  - "only when the error in the ChatGPT answer is obvious, users can identify the error. However, when the error is not readily verifiable or requires external IDE or documentation, users often fail to identify the incorrectness or underestimate the degree of error in the answer. ... From semi-structured interviews, it is apparent that polite language, articulated and text-book style answers, comprehensiveness, and affiliation in answers make completely wrong answers seem correct. We argue that these seemingly correct-looking answers are the most fatal. They can easily trick users into thinking that they are correct..."

Prompt Engineering

Concrete tips on how to write effective prompts

Exercise 3 – Rock, Paper, Scissors

You will be sent to a breakout room. Your task is to **ask ChatGPT to write code for a rock paper scissors problem**.

There are two parts. If you are able to make ChatGPT solve the first part, move on to part 2.

The prompt will be given in the chat. But can also be found here:

https://github.com/nuitrcs/code_chatgpt/

^ Response from DALL-E 3 for "Please generate an image of the game 'rock, paper, scissors' "

*The prompt is adapted (and mostly taken verbatim) from the second puzzle of 2022's Advent of Code. See here.

Exercise 3 – Rock, Paper, Scissors

Constraints:

- You may only write in natural language and cannot write anything in programming language.
- Do not provide the solution to ChatGPT. Instead, experiment with its reasoning as you give it **guiding prompts**.
- I used python for this, but you can use any other programming language.

Reminder

- Be clear in your instructions
- Ask the LLM to explain itself, and to describe its reasoning step by step
- Iteration is OK
- Ask it to explain how it understands certain concepts
- Ask it to summarize knowledge/results in a table or some other way.

Exercise 3– Rock, Paper, Scissors

PART 1

You will face an opponent in a game of Rock, Paper, Scissors, which will last for several rounds. You find a strategy guide for winning. The strategy guide consists of two columns. The first column is what your opponent is going to play: A for Rock, B for Paper, and C for Scissors. You are not sure what the second column means, but you reason it must be what you should play in response: X for Rock, Y for Paper, and Z for Scissors. Your total score is the sum of your scores for each round. The score for a single round is the score for the shape you selected (1 for Rock, 2 for Paper, and 3 for Scissors) plus the score for the outcome of the round (0 if you lost, 3 if the round was a draw, and 6 if you won). For example, suppose you were given the following strategy guide:

A Y

B X

C Z

This strategy guide predicts and recommends the following: In the first round, your opponent will choose Rock (A), and you should choose Paper (Y). This ends in a win for you with a score of 8 (2 because you chose Paper + 6 because you won). In the second round, your opponent will choose Paper (B), and you should choose Rock (X). This ends in a loss for you with a score of 1 (1 + 0). The third round is a draw with both players choosing Scissors, giving you a score of 3 + 3 = 6. In this example, if you were to follow the strategy guide, you would get a total score of 15 (8 + 1 + 6).

PART 2

Turns out the second column says how the round needs to end: X means you need to lose, Y means you need to end the round in a draw, and Z means you need to win. The total score is still calculated in the same way, but now you need to figure out what shape to choose so the round ends as indicated.

The example above now goes like this: In the first round, your opponent will choose Rock (A), and you need the round to end in a draw (Y), so you also choose Rock. This gives you a score of 1 + 3 = 4. In the second round, your opponent will choose Paper (B), and you choose Rock so you lose (X) with a score of 1 + 0 = 1. In the third round, you will defeat your opponent's Scissors with Rock for a score of 1 + 6 = 7.

Now that you're correctly decrypting the secret strategy guide, you would get a total score of 12.

Exercise 3 – Questions

- Did you tell ChatGPT how to play RPS?
- Did you check if it understood the rules correctly?
- Did you provide examples of what you wanted it to do?
- Did you check the code against unseen data?
- Did you ask ChatGPT to explain its reasoning?
- Did you ask ChatGPT to summarize data/functions in tables?
- Did you solve issues by going over the problem step by step?

Copilots

LLMs inside your IDEs

Lots of Copilots…

- GitHub Copilot (we will focus on this)
  : https://github.com/features/copilot
  - Supports lots of IDEs, including VScode, RStudio (preview)
- Jupyter AI : https://github.com/jupyterlab/jupyter-ai
- AWS CodeWhisperer
  : https://aws.amazon.com/codewhisperer/
- gptstudio : https://michelnivard.github.io/gptstudio/
  - More like a ChatGPT interface within RStudio
- Build your own using Hugging Face
  : https://huggingface.co/blog/personal-copilot
- (Microsoft also renamed Bing to "Copilot", which is now available on Windows desktop, but this is a conversation-style interface and not what we're referring to here!)

^ Response from DALL-E 3 for "Please generate an image showing that there are many LLM copilots available"

GitHub Copilot

- “AI pair programmer” that suggests code completions and turns natural language prompts into code
- Similarly to ChatGPT, you can use Copilot for code generation, explanation, translation, debugging, refactoring, test generation, and code reviews.
- Conveniently integrates with various IDEs such as VS Code
- Provides suggestions for many languages, but works better for Python, JavaScript, TypeScript, Ruby, Go, C#, and C++ (and less well for languages that are less represented in public repositories)
- There’s an ongoing lawsuit against GitHub Copilot, Microsoft, and OpenAI for allegedly violating open-source licenses.

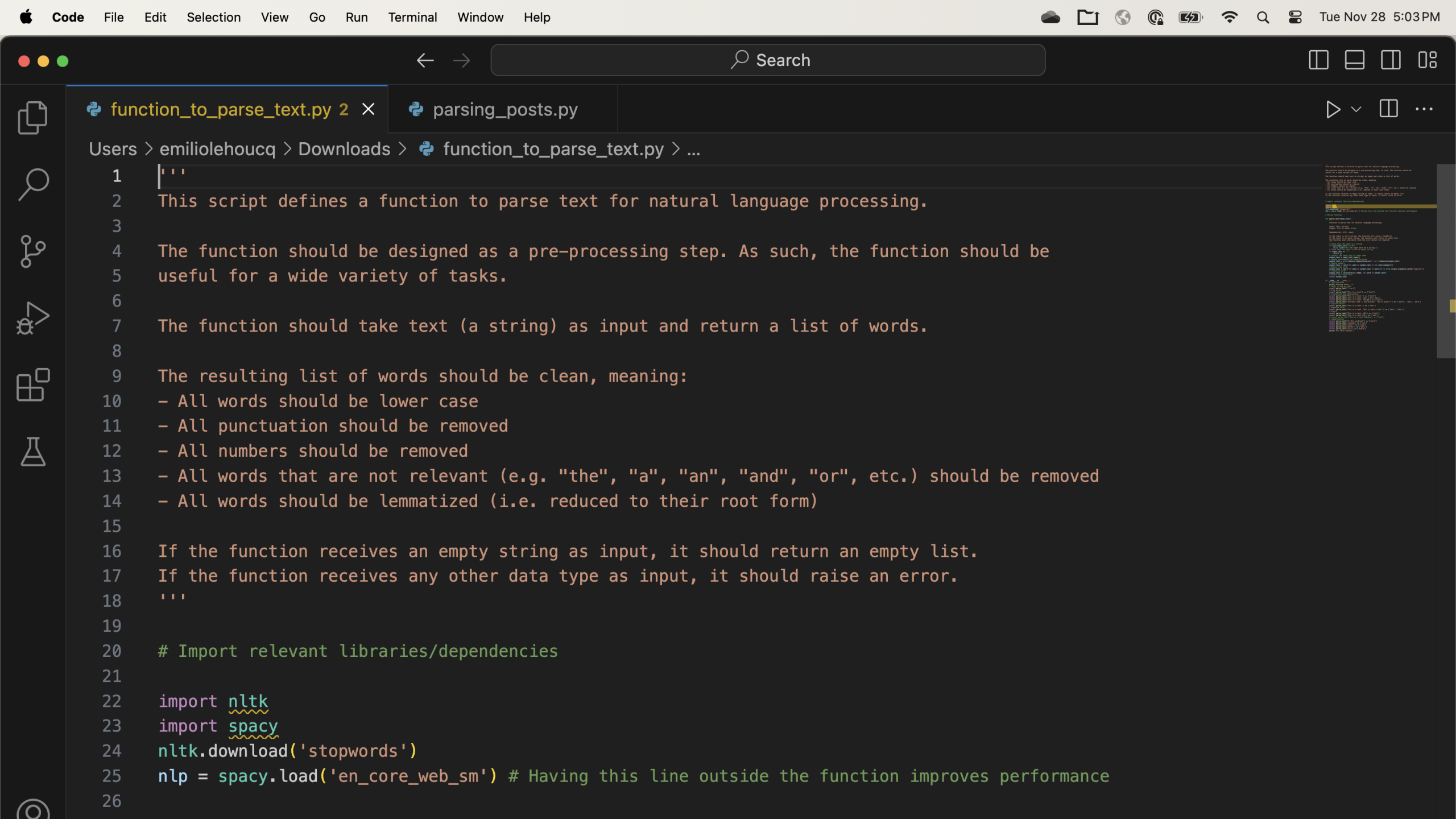
^ Response from DALL-E 3 for "Please generate an image of GitHub copilot helping a person code"

GitHub Copilot Best Practices

- Iterate over your prompts
- Allow Copilot to generate code in small steps
- Provide context
  - Explain the high-level goal at the beginning of your script.
  - Import the modules that you want to work with.
  - Write simple and specific instructions breaking down the logic and steps.
  - Give Copilot examples.
  - Keep a couple of tabs open in your IDE.
- Use good coding practices
  - Provide descriptive/meaningful variable and function names.
  - Follow a consistent/predictable style and pattern.
  - Structure your code into small functions.

GitHub Copilot in VS Code

- You need to install the GitHub Copilot extension and sign in
- You can also install the GitHub Copilot Chat extension
- More instructions and advanced functionality: https://code.visualstudio.com/docs/editor/github-copilot
- Inline suggestions
  - Start writing code and receive suggestions in gray faded text
  - Hover over the suggestion or
    - Tab to accept the suggestion
    - Cmd + right arrow to accept part of the suggestion
    - Option +] to see next suggestion(s)
- Chat
  - You can access the chat by clicking in the message icon in the activity (left side) bar or using ^ + cmd + I
  - Cmd + I to use inline in the script (esc to leave)


Code File Edit Selection View Go Run Terminal Window Help
Tue Nov 28 5:03 PM
Search
function_to_parse_text.py 2
parsing_posts.py
Users > emiliolehoucq > Downloads > function_to_parse_text.py > ...
'''
This script defines a function to parse text for natural language processing.
The function should be designed as a pre-processing step. As such, the function should be
useful for a wide variety of tasks.
The function should take text (a string) as input and return a list of words.
The resulting list of words should be clean, meaning:
- All words should be lower case
- All punctuation should be removed
- All numbers should be removed
- All words that are not relevant (e.g. "the", "a", "an", "and", "or", etc.) should be removed
- All words should be lemmatized (i.e. reduced to their root form)
If the function receives an empty string as input, it should return an empty list.
If the function receives any other data type as input, it should raise an error.
'''
# Import relevant libraries/dependencies
import nltk
import spacy
nltk.download('stopwords')
nlp = spacy.load('en_core_web_sm') # Having this line outside the function improves performance

Running Local LLMs

LLMs can run on your computer

Why + How to do this?

- Running locally means you aren't sharing your data, e.g., with OpenAI, Google, Microsoft (so less of a security/IP concern)
- You can potentially try many different LLMs to see with is best for you.
- LM Studio makes this easy! Though your milage may vary depending on your computer hardware and which model you choose.
- Here's what HuggingFace says about the best LLMs for coding.
    - I've had decent luck with the Llama-based models

^ Response from DALL-E 3 for "Please generate an image of a person using their laptop where the laptop is not connected to the cloud"

The LM Studio interface